

Werkzeuge für das wissenschaftliche Arbeiten

Python for Machine Learning and Data Science

Abgabe: 15.12.2023

Contents

| | |
|--------------------------|----------|
| 1 Projektaufgabe! | 1 |
| 1.1 Einleitung | 1 |
| 1.2 Aufbau | 1 |
| 1.3 Methoden | 2 |
| 2 Abgabe | 2 |

1 Projektaufgabe!

In dieser Aufgabe beschäftigen wir uns mit Objektorientierung in Python. Der Fokus liegt auf der Implementierung einer Klasse, dabei nutzen wir insbesondere auch Magic Methods.

1.1 Einleitung

Ein Datensatz besteht aus mehreren Daten, ein einzelnes Datum wird durch ein Objekt der Klasse `DataSetItem` repräsentiert. Jedes Datum hat einen Namen (Zeichenkette), eine ID (Zahl) und beliebigen Inhalt.

Mehrere Datenobjekte vom Typ `DataSetItem` sollen in einem Datensatz zusammengefasst werden. Es gibt eine Klasse `DataSetInterface`, die die Schnittstelle definiert und Operationen jedes Datensatzes angibt. Bisher fehlt aber noch die Implementierung eines Datensatzes mit allen Operationen.

Implementieren Sie eine Klasse `DataSet` als Unterklasse von `DataSetInterface`.

1.2 Aufbau

Es gibt drei Dateien: `dataset.py`, `main.py` und `implementation.py`.

- In `dataset.py` befinden sich die Klassen `DataSetInterface` und `DataSetItem`.
- In `implementation.py` muss die Klasse `DataSet` implementiert werden.

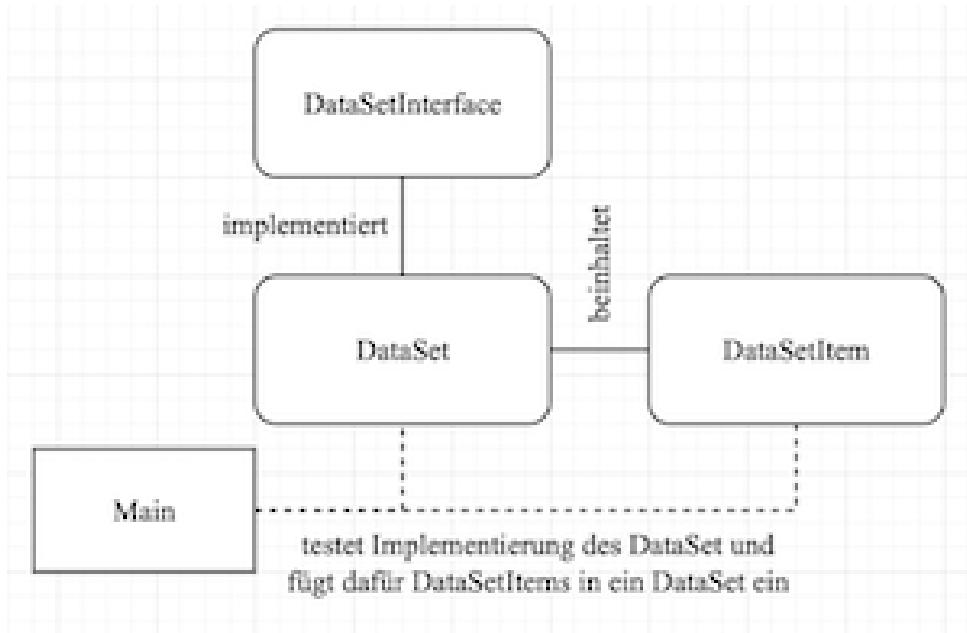


Figure 1: Darstellung der Klassenbeziehungen.

- `main.py` nutzt die Klassen `DataSet` und `DataSetItem` und testet die Schnittstelle und Operationen.

1.3 Methoden

Folgende Methoden sind in der Klasse `DataSet` zu implementieren (genaue Spezifikation in `dataset.py`):

- `__setitem__(self, name, id_content)` – Hinzufügen eines Datums.
- `__iadd__(self, item)` – Hinzufügen eines `DataSetItem`.
- `__delitem__(self, name)` – Löschen eines Datums nach Name.
- `__contains__(self, name)` – Prüfen, ob ein Datum vorhanden ist.
- `__getitem__(self, name)` – Abrufen eines Datums über Name.
- `__and__(self, dataset)` – Schnittmenge zweier Datensätze zurückgeben.
- `__or__(self, dataset)` – Vereinigungen zweier Datensätze zurückgeben.
- `__iter__(self)` – Iteration über alle Daten.
- `filtered_iterate(self, filter)` – Gefilterte Iteration mittels Lambda-Funktion.
- `__len__(self)` – Anzahl der Daten abrufen.

2 Abgabe

Programmieren Sie die Klasse `DataSet` in der Datei `implementation.py` zur Lösung der oben beschriebenen Aufgabe. Sie können auch direkt auf Ihrem Computer programmieren. Das VPL nutzt denselben Code, wobei `main.py` zusätzliche Testfälle enthält.